Action Recognition from 3D Skeleton Sequences using Deep Networks on Lie Group Features

Manel Rhif, Hazem Wannous, Imed Riadh

Action Recognition from 3D Skeleton Sequences using Deep Networks on Lie Group Features

Manel Rhif*, Hazem Wannous* and Imed Riadh Farah†
*Univ. Lille, CNRS, Centrale Lille, IMT Lille Douai,
UMR 9189 - CRIStAL - Centre de Recherche en Informatique
Signal et Automatique de Lille, F-59000 Lille, France
†University of Manouba, RIADI Laboratory ENSI, Tunisia

***Abstract*—This paper addresses the problem of human action recognition from sequences of 3D skeleton data. For this purpose, we combine a deep learning network with geometric features extracted from data lie on a non-Euclidean space, which have been recently shown to be very effective to capture the geometric structure of the human pose. In particular, our approach claims to incorporate the intrinsic nature of the data characterized by Lie Group into deep neural networks and to learn more adequate geometric features for 3D action recognition problem. First, geometric features are extracted from 3D joints of skeleton sequences using the Lie group representation. Then, the network model is built from stacked units of 1-dimensional CNN across the temporal domain. Finally, CNN-features are then used to train an LSTM layer to model dependencies in the temporal domain, and to perform the action recognition.**

The experimental evaluation is performed on three public datasets containing various challenges: UT-Kinect, Florence 3D-Action and MSR-Action 3D. Results reveal that our approach achieves most of the state-of-the-art performance.

I. Introduction

Human action recognition is a hot topic in the computer vision and pattern recognition fields. It is used for several applications related to machine learning techniques and human behavior as robotics, intelligent monitoring, human-computer interaction, virtual reality and video games, among others. The similarity of action due to the complexity of human movement and the variety of the same actions performed by distinct subjects are the fundamental challenges of actions classification.

In recent years, many approaches dealing with human action and activity recognition from depth sensors have received growing attention. These approaches can be categorized into 3D skeleton approaches, depth-based approaches and hydride approaches. In this paper, we focus on 3D skeleton data because it gives richer information about human kinematics. 3D skeleton data record the trajectories of human body joints and undergo less intra-class invariance compared to RGB and depth [28]. It is robust to illumination change and invariant to camera views. However, these skeletons are often noisy due to the difficulty in localizing body-parts, self-occlusions and sensor range errors [11]. Variety of techniques reformulating computer vision problems over non-Euclidean spaces, such as Riemannian manifolds, have received growing attention [20], [23], [22], [4], [5], [1]. In this paper, we consider such a manifold representation which considers the geometry of space, and more particularly focus on Lie Group features [23], [22]. Lie group is one of the manifold-based approaches that achieved the best result on several public datasets [23], [22], [1]. The main constraint of its representation is the misalignment of sequences in time and that makes distance metric inexact. Dynamic Time Warping (DTW) is the most common way to solve this problem. This step cost supplemental time. In addition, features are extracted per skeleton and then stacked, which makes this representation extremely high-dimensional. Despite the effectiveness of Lie group presentation, it is computationally expensive and extracts only spatial information from skeleton joints. Hence, temporal information needs to be exploited to express the dynamics of human motion [7], [10].

The Convolutional Neural Networks (CNNs) are deep neural network models, which have obtained state-of-the-art results in many tasks due to their ability to act as translation-invariant features extractors. They create a hierarchy of features, progressively more abstract, thanks to the staked convolutional operators. CNNs nowadays have achieved great success in image classification and video action recognition [12], [3]. On the other hand, the Long Short Term Memory networks (LSTMs) which are a special kind of the Recurrent Neural Networks (RNN) [9], [28], [15], [6], [10], are capable of learning long-term dependencies in time series problems like action recognition. However, it is still difficult for LSTMs to memorize the information of entire sequences with many time steps [26] and to extract high-level features [17]. The combination of CNNs and LSTMs has already been proposed in literature domains (speech, text and image recognition).

This work has been motivated firstly by the powerful capability of deep neural networks in learning compact and discriminative representations for images and videos, and second by the successful use of traditional manifold-based analysis in many computer vision applications. In particular, we build a deep neural network architecture performing action learning on Lie Groups. The architecture of our proposed solution captures time dependencies on more progressively abstract features extracted by convolutional operations from the geometric representation offered by Lie Groups.

The rest of this paper is organized as follows. Section 2 presents the related work on skeleton-based action recognition. In Section 3, our approach is described and then Lie group

representation and proposed deep network architecture are introduced. Section 4 presents the experimental results and finally Section 5 concludes the paper.

II. Related Works

3D action recognition has been a widely explored topic in computer vision. We will focus then in reviewing the works we consider relevant to two main categories: handcrafted geometric and deep learning methods based on skeleton data.

a) Handcrafted geometric approaches: Before the recent advent of deep learning techniques, most of the works focused on the design of a traditional chain, including handcrafted feature extraction from motion, dynamic modeling and classification. The first type of methods performed action recognition from direct measures of 3D joint parameters of the human body. Indeed, Wang et al. [24] developed an algorithm based on key pose mining of each action, which is considered as a set of ordered poses which is required to be close but not necessarily adjacent in the action sequences. The action classification was then performed by matching into the motif of each class and taking the maximum of matching score. Luvizon et al. [16] proposed a new framework to extract spatial and temporal features from subgroups of joints. Then, they aggregated features using k-means classifier, and finally, combined them using a metric learning method.

Besides, many other approaches have been recently proposed to exploit the differential geometry to represent skeleton data, so as to consider the non-linear nature of human motion. In [22], [23], the authors represented each skeleton as one element on the Lie Group, and the action sequence corresponded thus to a curve on this manifold. To handle rate variability among curves, DTW was employed to temporally align the curves. Finally, Fourier temporal pyramid representation is applied before a linear classification by SVM. The manifold assumption was computationally expensive, that is why [11] tried a new representation of data. They proposed a method catching the maximum of relationships between skeleton joints. They proposed two kernel tensor representations; Sequence Compatibility Kernel (SCK) and Dynamics Compatibility Kernel (DCK). SCK captured the spatiotemporal compatibility of joints in one sequence against those in the other. DCK explicitly modeled the action dynamics of a sequence. Tensors formed from these kernels were used to train an SVM. Slama et al. [20], [19] exploited no-Euclidean geometric properties to express the time series of skeletons as one point on a Grassmann manifold, where the classification is performed benefiting from the Riemannian geometry of this manifold. Similarly, Devanne et al. [4] extended this idea to represent a spatiotemporal motion characterized by full human skeleton trajectory. These motion trajectories are extracted from 3D joints and expressed in $\mathbb{R}^{60}$. The action recognition is performed using a K-NN classifier using geodesic distances obtained in open curve shape space.

b) Deep learning approaches: After the recent progress in deep learning techniques, many applications of computer vision field, including action recognition, have shown a change of paradigm. Veeriah et al. [21] proposed a differential Recurrent Neural Network (dRNN) which represented a gating scheme for the LSTM. Their dRNN emphasized the change in information gain caused by the salient motions between the successive frames. Instead of using the whole skeleton sequence, Du et al. [6] divided it into five parts according to the human physical structure. Then, they fed each part separately into bidirectional RNNs/LSTMs. As the number of layers increased, the representations extracted by the subnets were hierarchically fused to build a higher-level representation. The final representation of the skeleton sequences were fed into a single-layer perceptron, and the temporally accumulated output of the perceptron formed the final decision. Liu et al. [15] proposed a spatiotemporal LSTM (ST-LSTM) to explicitly model the dependencies between the joints and apply recurrent analysis over spatial and temporal domains. Besides, they introduced a trust gate mechanism to make LSTM robust to noisy input data. Zhang et al. [28] investigated a set of simple geometric features using 3-layer LSTMs, and showed that using joint-line distances as input requires less data for training. Lee et al. [13] proposed a new representation of data and they used a Temporal Sliding LSTM (TS-LSTM). In the first step, they transformed the coordinate system of all skeleton using translation, scale and rotation. Then, they extracted salient motion from the new data representation. Finally, they used a TS-LSTM network which calculated the average of multiple parts – short-term, medium term and long term – to get final features.

III. Approach

For the problem of human action recognition in the 3D joint space, we propose a deep network architecture, which we refer to as the Long-term Recurrent Convolutional Network on Lie Groups (LRCNLG), to learn the Lie group representations of skeletal data. The network model is then built from stacked units of 1-dimensional CNN across the temporal domain. Then we use the CNN-features to train a LSTM model to recognize action categories. The architecture of the deep neural network of our approach is presented in Figure 1.

A. Lie Group Representation for 3D Skeletal Data

Lie group is a topological group and a smoothed manifold presented as a vector space [8]. As presented by Vemulapalli et al. [22], the human action can be represented by the movement of the rigid body part. The relative geometry T between two body parts can be described using the 3D rotation R and translation d required to take one body part to the position and orientation of the other. Mathematically, the relative geometry between body parts is presented as a point in Lie group space ($SE(3) \times ... \times SE(3)$), and the relative geometry between all pairs of body parts is then presented as a curve in $SE(3) \times ... \times SE(3)$. However, the classification of the curves in $SE(3) \times ... \times SE(3)$ into different action categories is a difficult task due to the non-Euclidean nature of the space. That's why the authors in [22], [23] map between Lie Group and Lie Algebra as presented in the equation (1).

$$Lexp_G(u) = e^u, Llog_G(g) = log(g) \quad (1)$$

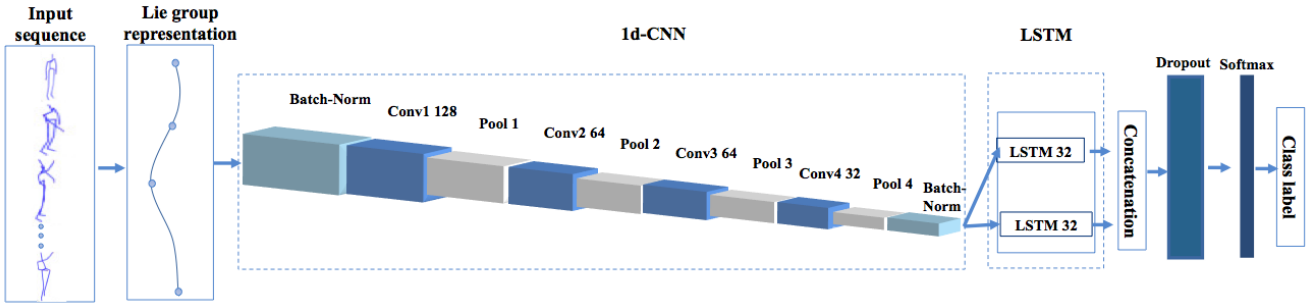

Fig. 1. The architecture of our LRCNLG model. The network model is built from stacked units of 1-dimensional CNN on Lie Group points across the temporal domain

where G represents the Lie Group, g its Lie Algebra, u the tangent space vector, e and log respectively represent the usual matrix exponential and logarithm. The Lie Algebra curve from Lie Group is given by an $6M(M-1)$ vector, where M is the number of body parts. The action is then represented as temporal evolution of d-dimensional vector.

B. Our LRCNLG model

To learn the features extracted from Lie Groups, we develop a LRCNLG model. The properties of our model follow those of classical CNN and LSTM ones. The LRCNLG model, trainable with stochastic gradient descent (forward and backward), takes as input Lie Group curves $(\zeta_1(t)...\zeta_n(t)$ where t=0,1,....T). This input is presented as a 3D tensor for variable-length sequences of (m, f)-dimensional vectors where m represent the maximum length for sequences and f is the feature dimension. More precisely, the model is built from stacked units of 1-dimensional CNN across the temporal domain followed by LSTMs. The different layers of our model are described below.

First, a layer of Batch Normalization (BN) allows to normalize data and to act like a scale-warping step. It restricts the inputs to follow a normal distribution. Then, a convolution layer is applied across the temporal domain. It is responsible for applying a mathematical computation of discrete convolution on the input feature (X) as 1-dimensional filters (K) presented in the equation (3):

$$(k * X)_i = \sum_m X_{i-m} k_m \tag{2}$$

The 1-dimensional filter height is fixed to the feature size f and its width to the length m. The kernel size is flipped to obtain the commutative property of convolution operations which leads to less variation of valid values. There are three hyper-parameters which control the output size of convolution layer: *the kernel size* which controls the number of neurons in the convolution layer that connects to the same region of the input tensor (its size is fixed to 8 in our model), *the stride* which specifies how many positions apart a filter is moved across the input, and *the zero-padding* which represents the size of the extending border values outside with 0s. In this implementation, the stride is always equal to 1 and there is no zero-padding performed. After sliding the filter over all the locations, we use a Rectified Linear Activation (ReLu) to the output in order to introduce nonlinearities and to leaves the size of the volume unchanged into the model. It is evaluated to 0 for negative inputs, and positive values remain untouched. ReLUs smooth approximations to the sum of many logistic units and they produce sparse activity vectors. Below is the equation of the ReLu function:

$$y = max\{0, \sum_{i=0}^{n} w_i x_i + x_0\} \tag{3}$$

The last layer in the CNN part of our model is a max-pooling one, which performs a down-sampling operation along the spatial dimensions. The 1-dimensional max-pooling partitions the input tensor data into 1D sub-tensors along the dimension, selects an element with the maximal numeric value in each sub-tensor, and transforms the input tensor to the output tensor by replacing each sub-tensor with its maximum element. In our model, we use four convolution and pooling layers.

Concerning the RNN part of our model, a typical LSTM is adopted to get the contextual dependency in the temporal domain. In our LRCNLG model, We concatenate two LSTM layers, and a dropout layer is then introduced to avoid overfitting problem. Finally, we used dense layer with a softmax function to transform the output codes to probability values of class labels. A dense layer represents a matrix vector multiplication. The values in the matrix are the trainable parameters which get updated during backpropagation. As a result we get a c-dimensional vector as output with c classes.

IV. Experimental Results

This section summarizes all obtained results and provides an analysis of the performances of our proposed approach tested on three public 3D action recognition datasets – UTKinect-Action [27], Florence 3D-Action [18] and MSR-Action 3D [14] – and compared with state-of-the-art methods.

A. Implementation Settings

For the feature extraction, we use the code of [22], [23] to represent action sequence as a Lie Group curve. Each action sequence is then represented as a matrix of 6*M(M-1) lines and f columns, where M is the number of joints per skeleton (20 for MSR-Action 3D and UTKinect, 15 for Florence 3D Action) and

f a fixed number of the sequence length (normalized as in [22], [23] to 76, 74 and 35 respectively for the 3 datasets). We used an initial learning rate of 0.01, stochastic gradient descent with nesterov acceleration with a momentum of 0.9 and a dropout with rate 0.5 after all activation layers to prevent overfitting. We used the Keras deep learning framework with a TensorFlow backend [2] on a laptop with an i5-2320 (3.00GHz) without GPU.

B. Human Action Datasets

UTKinect-Action [27]: In this dataset, skeletons were extracted using a single stationary Kinect with Microsoft SDK. Totally, there are 199 sequences whose length varies from 5 to 120 frames. These sequences were recorded from 10 subjects: 9 males, 1 female; one was left-handed. Each subject performed each action twice. There are 10 actions. UTKinect gives information regarding 20 joint locations for the 3D skeleton data. It is a challenging dataset due to the different variations of the views among records and its high intra-class variability. Furthermore, the human object interactions and the absence of some body parts in the view cause different occlusions.

Florence 3D-Action [18]: This dataset was collected using a stationary Kinect sensor at the University of Florence. It includes 9 actions, each one is performed by 10 subjects several times leading to a total of 215 sequences. The sequences were acquired using the OpenNI SDK, skeletons are represented by 15 joints. The main challenges of this dataset are the similarity between actions, the human object interactions and the diverse ways of representing the same action. The later generates a high intra-class variation, for example, the same action is performed using left hand in some sequences and right hand in other sequences.

MSR-Action 3D [14]: This dataset was captured using a depth sensor similar to Kinect at Microsoft research. It contains totally 567 sequences consisting of 20 actions presented by 10 persons facing the camera. Each action is performed 2 or 3 times. It is the most common dataset for 3D action recognition. It contains 20 joints to represent skeleton. Contrary to UTKinect and Florence, there is no interaction between persons and objects for all actions. Actions are taken in the context of gaming which induces many variations of the motions of the legs, arms, torso and their combination. In our experimentations, we use only 557 sequences because there are 10 actions that contain missing information and are erroneous [25]. This dataset is challenging due to the strong inter-class similarity and the speed variations for the execution of actions.

C. Action Recognition Analysis

To fairly compare our approach with the state-of-the-art methods, we follow the same experimental setup and evaluation protocol presented in these methods. Then, we present our result separately for each dataset.

1) Florence 3D-Action: The two most used protocols for this dataset are Leave-one-subject-out-cross-validation (LOOCV) [16] and cross-validation protocol [22], [11] for which half of the subjects is used for training and the other half is used for testing. To evaluate our proposed method, we choose to use both of protocols to be more extensive. As summarized in Table I, our approach achieves the best accuracy of 95.37% using LOOCV protocol. However, using the cross-validation protocol, it achieves an accuracy of 92.55%, just below the best one which used a kernel Linearization and reported in [11]. They used a combination of two methods the SCK and DCK kernels making their approach complex. As shown from the confusion matrix in Figure 2, despite the high accuracies for most actions, there are little confusions, like between *answer phone* and *drink from a bottle* as they have visually the same trajectory, or between *read watch* and *clap hand*, where both need two hands to perform the action.

Method	Protocol	Accuracy (%)
Lie groups [22]	cross validation	90.88
Kernel Linearization [11]	cross validation	95.23
shape analysis [4]	LOOCV	87.04
Mining key pose [24]	LOCCV	92.25
Feature combination [16]	LOOCV	94.39
Ours	LOOCV	**95.37**
Ours	cross validation	92.55

TABLE I
RESULTS OBTAINED ON FLORENCE 3D-ACTION USING TWO DIFFERENT PROTOCOLS.

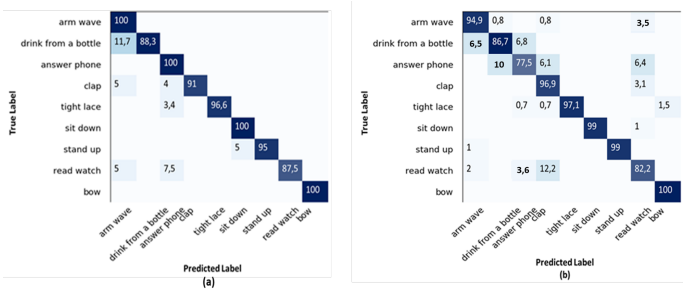

Fig. 2. Confusion matrix for our approach on Florence 3D Action using (a) LOOCV protocol and (b) cross-validation protocol

2) MSR 3D-Action: There are two basic protocols used by the state-of-the-art methods for this dataset: cross-validation protocol and 5-fold protocol which uses subjects 1, 3, 5, 7, 9 for the training while the subjects 2, 4, 6, 8, 10 are used for the testing. In our experiments, we choose to use cross-validation protocol because using the average between 10 splits is more precise than using just one split.

MSR Action 3D is a challenging database that is why generally authors divide it into three subsets and then calculate the average to get the accuracy of this dataset. Action presented in the first subset (AS1) are *Horizontal arm wave, Hammer, Forward punch, High throw, Hand clap, Bend, Tennis serve, Pickup and throw*. *High arm wave, Hand catch, Draw x, Draw tick, Draw circle, Two hand wave, Forward kick, Side boxing* are the actions in subset two (AS2). Finally, actions of the subset three (AS3) are *High Throw, Forward kick, Side kick, Jogging, Tennis swing, Tennis serve, Golf swing, Pickup and*

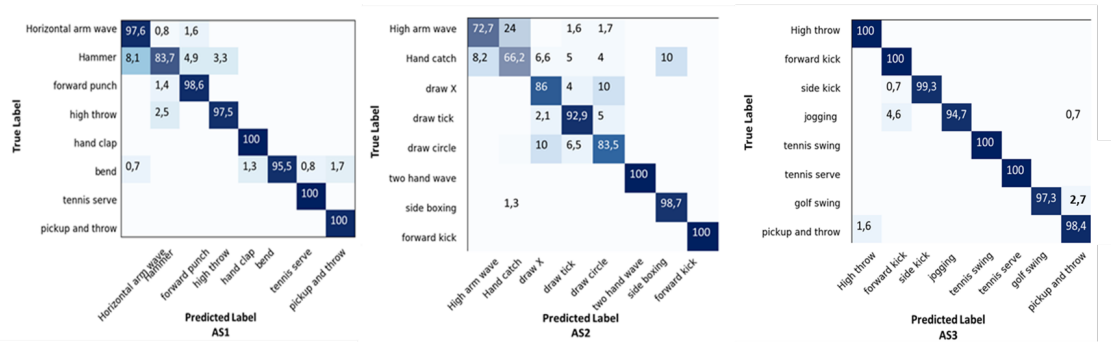

Fig. 3. Confusion matrix for our approach on MSR Action 3D dataset: (left) AS1, (center) AS2 (right) AS3.

Method	Protocol	Accuracy (%)
Lie groups [22]	cross validation	93.52
Kernel Linearization [11]	cross validation	93.96
Grassmann Manifold [20]	cross validation	91.21
dRNN [21]	cross validation	92.03
shape analysis [4]	cross validation	92.1
Mining key pose [24]	5-fold	94.40
Feature combination [16]	5-fold	97.1
HBRNN-L[6]	5-fold	94.49
TS-LSTM [13]	5-fold	97.22
Ours	cross validation	**94.27**

TABLE II
OBTAINED RESULTS ON MSR 3D ACTION DATASET

Method	Protocol	Accuracy (%)
Feature combination [16]	LOOCV	98
St-LSTM+trust gate [6]	LOOCV	97
Grassmann Manifold [20]	LOOCV	88.5
JLd+RNN [28]	cross validation	95.96
TS-LSTM [13]	cross validation	96.97
Lie groups [22]	cross validation	97.08
Kernel Linearization [11]	cross validation	98.2
Ours	LOOCV	**98.5**
Ours	cross validation	96.68

TABLE III
OBTAINED RESULTS ON UTKINECT-ACTION DATASET USING TWO DIFFERENT PROTOCOLS.

throw. Table II presents the accuracies of our approach obtained on this dataset, compared to the most relevant state-of-the-art methods using different protocols.

As shown, our approach achieve the best accuracy (94.27%) using the cross validation protocol, with 96.64% in AS1, 87.52% in AS2 and 98.71% in AS3. Our accuracy is however, outperformed by [13], [16] where they used only the 5-fold protocol. The confusion matrix presented in Figure 3 gives us more information on each action. Indeed, in AS1, we get the high accuracies for most actions (e.g. *Pickup and Throw*, *Tennis serve*), but *Hammer* is the worst classified action with 83.7% which has been confused with a similar one namely *Horizontal arm wave*. In AS2, there are many actions that are not perfectly classified e.g., *Hand catch* with 66.2% and *High arm wave* with 72.7%. In AS3, all actions are practically well classified with an accuracy over 95%.

3) UTKinect-Action: In this dataset, we kept the same protocols used for Florence 3D-Action dataset. As presented in Table III, our approach achieves the best performance (98.5%) using the LOOCV protocol, compared to all other skeleton-based approaches. As presented in Figure 4-a, all actions have been correctly classified (100%) expect *throw* (85%). These results prove the robustness of our approach against different challenges of the database. Using cross-validation protocol, we get a performance comparable to [28] and [13] with an accuracy of 96.68%, but stills 0.4% less than [22].

If we compare accuracies obtained by our approach to those obtained by the original Lie group representation [22], we note

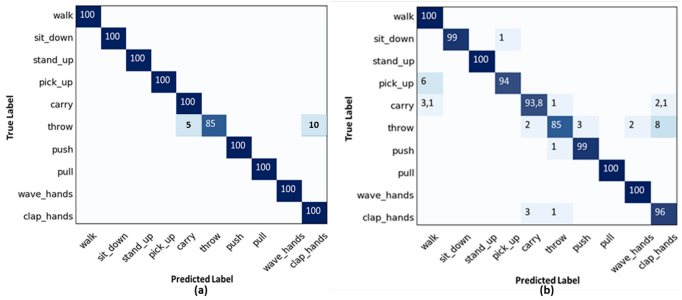

Fig. 4. UTKinect Confusion matrix using (a) LOOCV protocol and (b) cross-validation protocol

that our approach outperforms them for all actions, except for three ones: *Pick up, Carry and Clap hands*. These actions involve an interaction with objects and variations in viewpoint which make the classification harder. Note that results reported in [22] using the LOOCV protocol have showed a lower accuracy of 96.5% compared to our approach. It can be concluded that the LOOCV protocol is the more suitable one for deep learning models as a sufficient quantity of data sequences is needed for training subset, contrary to the case where only half of data is used for training.

D. Effectiveness of Geometric Lie Group Features

To assess the effectiveness of Lie Group features combined with our deep network model, we have repeated all experiments using our model applied directly on the 3D coordinates of

Dataset	Method	Protocol	Accuracy (%)
UTKinect	LRCN	LOOCV	95.50
		cross validation	89.45
	LRCNLG	LOOCV	**98.50**
		cross validation	96.68
Florence	LRCN	LOOCV	89.43
		cross validation	85.88
	LRCNLG	LOOCV	**95.37**
		cross validation	92.55
MSR Action 3D	LRCN	cross validation	92.4
	LRCNLG	cross validation	**94.27**

TABLE IV
OBTAINED ACCURACIES BY OUR MODEL ON LIE GROUP REPRESENTATION (LRCNLG) COMPARED TO THE SAME MODEL ON SKELETON COORDINATES IN $\mathbb{R}^3$ (LRCN).

skeleton data represented in $\mathbb{R}^3$ Euclidean space. We note here that we have performed a view normalization prior to feeding the 3D coordinates into the model. The results obtained by LRCNLG model are superior to those obtained by the model on $\mathbb{R}^3$ for all the datasets and using the two protocols. Table IV summarizes all the results obtained by our LRCNLG model on Lie Group features compared to those obtained by the same model, we call LRCN, on skeleton coordinates in $\mathbb{R}^3$, using two different protocols tested on all datasets. This can be explained by the fact that our LRCNLG model learns more adequate geometric features than traditional ones [22] but also than that with original Euclidean coordinates, for 3D action recognition task.

V. CONCLUSION

This paper addressed the problem of human action recognition in the 3D joint space. We introduced a novel framework, in which we built a neural network model to deeply learn geometric feature captured by Lie Group representations of skeletal data. We then formulated our learning architecture as a hierarchy of spatial CNN features extracted from each Lie Group point, followed by the LSTMs to model dependencies in the Lie Group curve representing the action sequence on a Riemannian manifold. Experimental results on 3 human action datasets consistently demonstrated the effectiveness of the proposed approach.

REFERENCES

[1] R. Anirudh, P. Turaga, J. Su, and A. Srivastava, "Elastic functional coding of riemannian trajectories," *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, vol. 39, no. 5, pp. 922–936, 2017.

[2] F. Chollet, 2015. [Online]. Available: https://github.com/fchollet/

[3] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 3642–3649.

[4] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. D. Bimbo, "3-d human action recognition by shape analysis of motion trajectories on riemannian manifold," *IEEE Trans. Cybernetics*, vol. 45, no. 7, pp. 1340–1352, 2015.

[5] M. Devanne, H. Wannous, P. Pala, S. Berretti, M. Daoudi, and A. D. Bimbo, "Combined shape analysis of human poses and motion units for action segmentation and recognition," *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, vol. 07, pp. 1–6, 2015.

[6] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *IEEE conference on computer vision and pattern recognition (CVPR)*, 2015, pp. 1110–1118.

[7] A. Gaidon, Z. Harchaoui, and C. Schmid, "Temporal localization of actions with actoms," *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, vol. 35, no. 11, pp. 2782–2795, 2013.

[8] B. Hall, *Lie groups, Lie algebras, and representations: an elementary introduction*. Springer, 2015, vol. 222.

[9] Z. Huang, C. Wan, T. Probst, and L. Van Gool, "Deep learning on lie groups for skeleton-based action recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE computer Society, 2017, pp. 6099–6108.

[10] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3d action recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[11] P. Koniusz, A. Cherian, and F. Porikli, "Tensor representations via kernel linearization for action recognition from 3d skeletons," in *European Conference on Computer Vision*. Springer, 2016, pp. 37–53.

[12] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[13] I. Lee, D. Kim, S. Kang, and S. Lee, "Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 1012–1020.

[14] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2010, pp. 9–14.

[15] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal lstm with trust gates for 3d human action recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 816–833.

[16] D. C. Luvizon, H. Tabia, and D. Picard, "Learning features combination for human action recognition from skeleton sequences," *Pattern Recognition Letters*, 2017.

[17] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4580–4584.

[18] L. Seidenari, V. Varano, S. Berretti, A. Bimbo, and P. Pala, "Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 479–485.

[19] R. Slama, H. Wannous, and M. Daoudi, "Grassmannian representation of motion depth for 3d human gesture and action recognition," in *22nd International Conference on Pattern Recognition (ICPR)*, 2014, p. 34993504.

[20] R. Slama, H. Wannous, M. Daoudi, and A. Srivastava, "Accurate 3d action recognition using learning on the grassmann manifold," *Pattern Recognition*, vol. 48, no. 2, pp. 556 – 567, 2015.

[21] V. Veeriah, N. Zhuang, and G.-J. Qi, "Differential recurrent neural networks for action recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4041–4049.

[22] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2014, pp. 588–595.

[23] R. Vemulapalli and R. Chellappa, "Rolling rotations for recognizing human actions from 3d skeletal data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4471–4479.

[24] C. Wang, Y. Wang, and A. L. Yuille, "Mining 3d key-pose-motifs for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2639–2647.

[25] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),*, 2012, pp. 1290–1297.

[26] J. Weston, S. Chopra, and A. Bordes, "Memory networks," *International Conference on Learning Representations (ICLR)*.

[27] L. Xia, C.-C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 20–27.

[28] S. Zhang, X. Liu, and J. Xiao, "On geometric features for skeleton-based action recognition using multilayer lstm networks," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 148–157.